# Evidential Distributed Dynamic Map for Cooperative Perception in VANets

Nicole El Zoghby[1], Véronique Cherfaoui[1] and Thierry Denoeux[1]

***Abstract*— In this paper, we present a distributed approach to build a dynamic map in the context of VANets (Vehicular Ad hoc Networks). It is based on the principle of cooperative perception where vehicles work as a team in order to extend their field of view. Each vehicle is equipped with sensors allowing it to detect its environment and to build its map, denoted by local map. It receives messages from other vehicles containing mobile objects detected in their surroundings. The algorithm of distributed dynamic map builds a map of the dynamic environment including objects in the sensor's field of view as well as those sent by other vehicles. This algorithm is developed under the belief functions framework. The implementation of such an application is complex and needs many treatments: temporal and spatial alignment, object association, fusion of messages and data dissemination. This approach has been validated by simulation on scenario involving several vehicles in traffic situation.**

## I. INTRODUCTION

Nowadays, the research field of intelligent vehicles systems benefit from the emerging inter-vehicle communication to perform cooperation. Vehicles are equipped with sensors for surroundings' detection, absolute localization system for localization and wifi antenna for broadcasting information. Different applications can be considered such as cooperative localization and perception.

Cooperative localization has been treated in several domains such as robots cooperation and intelligent vehicles applications. Cooperative localization approaches differ with the data exchanged in the network. We can distinguish two classes of approaches: in the first one, vehicle sends only its own detected data ([17], [13], [9], [18]), while in the second approach, it sends what it receives from others or it combines its own local data with received data ([12], [4]). The latter case is a distributed method where the same information can be combined several times. This phenomenon is called data incest and can be avoided with an appropriate treatment. A review on cooperative localization can be found in [4].

We present a distributed approach that consists of sending the result of data fusion of own vehicle detection with received messages. It is a cooperative perception approach where vehicles cooperate in order to extend their field of view (constrained by sensor's limitations and occultation) and to reduce the false alarms. The purpose of this method is not to improve the accuracy of objects localization but to extend vehicle's perception range by reinforcing confidence in objects existence. As in [3] and [15], we use the belief functions framework to manage uncertainty of object existence.

Recent research works have addressed cooperative perception in the robotics and intelligent vehicles fields. Merino et al. [14] have developed a cooperative perception system for heterogeneous multi-UAV. Their system considers heterogeneous sensors. They presented the system architecture and experimental results on automatic forest fire detection and localization. Li and Nashashibi [11] have presented a method of cooperative perception for augmented reality application. Their method was applied on an example with two vehicles. The idea is to transform the occulted part of the first vehicle to a perception of the rear vehicle based on the 3D perspective. The authors estimate the relative pose between two reference vehicles in order to have a common reference for vehicle perception, allowing them to transform perceptions of vehicles in 3D perspective. Their method has been tested on experimental vehicles. Raugh et al [16] have established a cooperative perception system. Car2X-based perception is modeled as a virtual sensor in order to integrate it into a high-level sensor data fusion architecture. Temporal and spatial alignments are performed to improve vehicles state accuracy. Their method is validated using experimental data. In [10], authors used cooperative perception for vehicle control on the road. The proposed cooperative systems had been implemented on self-driving vehicle and manned vehicles.

We consider the concept of Local Dynamic Map (LDM) developed in the Safespot project [2]. This map includes static and dynamic information in the surroundings of a vehicle. This information is updated periodically. The LDM is divided into four levels. We are interested in the fourth level including the mobile elements in the scene (ego-vehicle, other vehicles, pedestrians, trucks, ...). The aim is to exchange these maps (fourth level only) between vehicles in order to increase the field of view of each one of them. Each vehicle sends its current pose and the list of objects. We present in this paper the algorithm that allows combining the maps exchanged between vehicles. This algorithm is an extension of the distributed data fusion algorithm presented in [20]. It is developed under the belief functions framework. This framework seems appropriate for its ability to model uncertainties and to combine data using adapted operator that considers data dependency.

For sake of simplicity, in what follows, we replace the notion of local dynamic map by the dynamic map $DM$ since the vehicle will have two types of information: local and distributed.

[1] Heudiasyc, CNRS UMR 7253- Université de Technologie de Compiègne, France, nicole.el-zoghby@hds.utc.fr, veronique.cherfaoui@hds.utc.fr, thierry.denoeux@hds.utc.fr

This paper is organized as follows. The problem is stated in Section II. In Section III, we present our distributed dynamic map algorithm based on the belief functions framework. Implementation and results are presented in Section IV. Finally, Section VI concludes the paper.

## II. Problem Statement

### A. Dynamic Map and vehicle's state

We consider that each vehicle $V_j$ detects with its sensors a map denoted by $DM_j$. The vehicle map contains a list of objects $\{O_i\}_j$ (where $i$ is the identity of the object). Each vehicle has a confidence in the existence of the detected object, this confidence is represented by $m_i$.

The vehicle $V_j$ is ego-localized and knows its state represented by $X_{V_j} = (x_j, y_j, \varphi_j, v_j, P_j)$ where $x_j, y_j$ represent the vehicle's absolute position, $\varphi_j$ its orientation, $v_j$ the vehicle's velocity and $P_j$ represents the covariance matrix.

Objects $\{O_i\}_j$ are represented by different attributes in the ego vehicle reference frame such as:

- $p_i$, $v_i$ and $c_i$ position, velocity and class of object $i$,
- $\sigma_{p_i}$ and $\sigma_{v_i}$ uncertainties of position and velocity or the covariance matrix represented by $P_i$,
- $m_{c_i}$ the mass function that represents the degree of belonging to class $c_i$. This information depends on the type of the sensor used to detect the scene. The class mass function is only used in the association.
- $m_i$ the mass function of the object existence.

The aim is to evaluate the confidence in the objects detected in the scene. We are interested in objects' existence. The frame of discernment is: $\Omega = \{O, NO\}$ where "O" represents the objects (vehicle, truck, bus, ...) and "NO" non-objects (false alarms due to the detection of the road, sidewalk, ...). The confidence on the object existence is represented by mass functions as in the following:

$$m_i(O) = a, m_i(NO) = b, m_i(\Omega) = 1 - a - b. \tag{1}$$

The mass functions assign a degree of belief to the different parts of $\Omega$ where $m_(\Omega)$ represents the uncertainty. The computation of these mass functions is detailed in section IV-A.

### B. Local and distributed map

To implement a distributed algorithm, each vehicle $V_j$ must have two types of information: local and distributed. The local map, denoted by $DLM$, is what the vehicle detects with its own sensor and contains the mass it gives to the detected objects. The vehicle keeps this information and does not send it as it is. It combines it with the received messages in order to establish the distributed map. As for the distributed map, we distinguish between $DDM$ (Dynamic Distributed Map) and $DPM$ (Dynamic Public Map). The $DDM$ is a $DM$ updated with maps received from others. The $DPM$ is the result of the combination of the distributed map and the local map. The $DPM$ is sent over the network.

To exploit messages, each vehicle should transform received data in a global reference frame, execute a temporal alignment and associate the detected objects. Objects are associated using the algorithm developed in [19]. After these different steps, the vehicle can update its distributed map. This is possible in VANets thanks to the GPS pose and GPS common time.

## III. Distributed Dynamic Map algorithm

Algorithm 1 presents the Distributed Dynamic Map algorithm based on the distributed data fusion algorithm developed in [20]. This algorithm is applied by the receiver when it receives a message containing the sender's map. In this algorithm, the sender information is indexed by $s$ and the receiver information by $r$.

The principle of this approach is that when a vehicle receives a message, it updates its distributed map with the received map using the cautious rule [6]. Each vehicle can combine the same information many times as it is coming from independent sources. This is called data incest problem provoked by cycles of data dissemination. To avoid this problem, the idempotent cautious rule is used. It is also an associative, commutative rule and defines an order relationship on the weights. After that, it combines its distributed map with its local map by the Dempster's rule [5]. This rule is based on conjunctive operator and must be used when sources are independent (this is the case here). The result is used to update the public map. The vehicle sends its public map over the network. The convergence of this algorithm had been demonstrated in [7]. It was shown, in this latter paper, that this algorithm is robust to errors. This property is ensured by the fact that the received information has to be discounted before the combination. The algorithm is detailed in the following subsections.

**Algorithm 1:** Distributed Dynamic Map Algorithm

**Input:** Message from sender $DPM_s$;
**Output:**$DPM_r$;$DDM_r$ ;
**Update $DDM_r$ with message**
$DPM_s \leftarrow^{\alpha} DPM_s + S$: Discounting of the $DPM_s$ of the sender and adding the sender in the list of objects;
$\widehat{DPM}_s \leftarrow prediction(DPM_s)$;
$\widehat{DDM}_r \leftarrow prediction(DDM_r)$;
$DDM_r \leftarrow FusionCautious(\widehat{DPM}_s, \widehat{DDM}_r)$;
**Compute $DPM_r$ with Local Map**
$DLM_r \leftarrow local\ map\ acquisition()$;
$DPM_r \leftarrow FusionDempster(DDM_r, DLM_r)$;
Send $DPM_r$;

### A. Update the $DDM$ of receiver with message

**Discounting**: In order to ensure the algorithm convergence [7], each receiver has to discount the received data. This operation tends to give less confidence in received information. For this aim, the receiver applies a discounting on the sender $DPM_s$. This discounted map is denoted by ${}^{\alpha}DPM_s$. It also adds the sender $S$ to its map. The discounting is applied on the existence mass functions as follows:

$$\begin{array}{lcll} {}^{\alpha}\text{m(A)} & = & \alpha.\text{m(A)} & \text{where } A \neq \Omega, \\ {}^{\alpha}\text{m}(\Omega) & = & (1\text{-}\alpha)\text{+}\alpha.\text{m}(\Omega). & \end{array} \tag{2}$$

**Prediction**: A temporal alignment is needed before updating the receiver distributed map with the sender's one. A transmission delay is assumed in this approach. The temporal alignment is to predict these two distributed maps at time $t'$ of the treatment with a prediction model with constant velocity. The prediction function ($\widehat{DPM_s} \leftarrow$ prediction($DPM_s$), $\widehat{DDM_r} \leftarrow$ prediction($DDM_r$)) allows the maps prediction along these equations:

$$O \begin{cases} x(t') &= x(t) + v_x(t) * \Delta t, \\ y(t') &= y(t) + v_y(t) * \Delta t, \\ P(t') &= F.P(t).F^T + Q, \\ m_c^{(t')} &\leftarrow m_c^{(t)}, \\ v(t') &= v(t). \end{cases} \tag{3a}$$

$$m \left\{ \; m^{(t')} \leftarrow {}^{\alpha'} m^{(t)}. \right. \tag{3b}$$

where $\Delta t = t' - t$, $t'$ is the prediction time and $t$ is the time when the map is build, $F$ is the model matrix that relates the state at time $t'$ with the state at time $t$ and $Q$ is the covariance matrix. The existence mass functions of the objects are discounted in terms of time. The discounting factor is represented by $\alpha'$ where $\alpha' = exp^{(-\Delta t)}$.

**Fusion**: The predicted maps are associated using the association algorithm detailed in [19] that takes into account the objects position, velocity and class. Algorithm 2 presents the $FusionCautious$ function that allows considering three cases when fusing objects:

- if objects $(O_s, O_r)$ are associated, the receiver existence mass is updated with the cautious rule [6],

$$m_r^{(t)} = m_r^{(t)} \owedge m_s^{(t)}; \tag{4}$$

- if the sender has not detected an object already detected by the receiver ($O_r$ is not associated), the object existence mass function is discounted as in equation 2;
- if the sender has detected an object not detected by the receiver, this object is added to $DDM_r$.

**Algorithm 2:** FusionCautious

**Input:** $\widehat{DPM_s}, \widehat{DDM_r}$;
**Output:**$DDM_r$;
$Association(\widehat{DPM_s}, \widehat{DDM_r})$ ;
  For each associated objects $O_s$ and $O_r$
    $m_r^{(t)} \leftarrow m_r^{(t)} \owedge m_s^{(t)}$
  For each $O_r$ not associated
    $O_r \leftarrow^{\alpha} O_r$
  For each $O_s$ not associated
    $DDM_r \leftarrow DDM_r + O_s$

### B. Compute the receiver $DPM$ with its $DLM$

The dynamic public map $DPM_r$ is the result of combination of the distributed map ($DDM_r$) and local map ($DLM_r$) of the receiver. The $DLM_r$ is built by the receiver embedded sensors. This map is then combined to the distributed map. For this aim, both maps are associated before the update step. The result is saved in $DPM_r$ that will be sent to other vehicles. Algorithm 3 shows the three cases that should be taken into account for the objects' fusion:

**Algorithm 3:** FusionDempster

**Input:** $DLM_r$,$DDM_r$;
**Output:**$DPM_r$ ;
$DPM_r \leftarrow DDM_r$
$Association(DDM_r, DLM_r)$ ;
  For each associated objects $O_r$ and $O_{r_l}$
    $O_r \leftarrow O_{r_l}$
    $m_r^{(t)} = m_{r_l}^{(t)} \oplus m_r^{(t)}$
  For $O_{r_l}$ not associated
    $DPM_r \leftarrow DPM_r + O_{r_l}$
  For each $O_r$ not associated
    **if** $inFieldofView$ **then**
      $delete(O_r)$
**else**
  $keep(DPM_r)$

- if objects $(O_r, O_{r_l})$ are associated, $O_r$ is updated with $O_{r_l}$, where $O_{r_l}$ is the object detected by the receiver in its local map. The mass functions are combined in this case by the Dempster's rule [5] as follows:

$$m_r^{(t)} = m_{r_l}^{(t)} \oplus m_r^{(t)}; \tag{5}$$

- if $O_{r_l}$ is not associated, we keep it in the $DPM_r$;
- if the object $O_r$ of the distributed map $DDM_r$ is not associated and should be in the receiver's field of view, we delete it because it is considered as a false alarm. If it is not in the field of view, we add it to $DPM_r$.

Finally, $DPM_r$ is sent over the network.

We should note that the local map in this algorithm is injected once, because the objects are supposed to be tracked in time by intelligent sensors.

## IV. Implementation

The Distributed Dynamic Map algorithm have been validated on simulated data created by A. Houenou [8]. We have simulated several scenarios with different vehicles, some of them are equipped with an intelligent camera (field of view: $maxrange = 60m$ and $angle = 45°$), a localization system and a wifi antenna. Vehicles, equipped with a wifi antenna, share their distributed dynamic map. They move along a multi-lane road and detect their surroundings.

The intelligent camera is a tracking system that provides different types of information such as object's identity, its relative position $x$ and $y$ and its covariance matrix $P_x$, its relative velocity $v_x$ and $v_y$ and the covariance matrix $P_v$, object's age and class. This provided information allows computing mass functions and building the dynamic maps.

This algorithm needs experimental setup. In the following, the discounting factor is fixed to 0.8 empirically and it is the same for all the senders. It may depend on the number of hop, the confidence in sender. The association parameters are chosen as detailed in [19].

### A. Mass functions computations

The object's age provided by the sensor of vehicle $V_j$ presents the number of times the object $O_i$ has been detected. The object's existence mass function is computed in this manner:

$$\begin{array}{rcccc} m(O) & = & a & = & \beta.(1 - \mathrm{e}^{-k.age(O_i)}), \\ m(NO) & = & b & = & \beta.(\mathrm{e}^{-k.age(O_i)}), \\ m(\Omega) & = & 1-a-b & = & (1-\beta), \end{array} \quad (6)$$

where $\beta = 0.9$ represents the sensor's reliability and $k$ is a positive coefficient $k = 0.1$. The more the sensor detects the object, the more the existence mass function is enhanced.

The simulator does not provide the class mass function $m_c$. The class frame of discernment is $\Omega_c = \{V, NV\}$, where $V$ represents the fact that the object is a vehicle and $NV$ not a vehicle. We assign a mass 0.9 to $\{V\}$ or $\{NV\}$ depending on the sensor's decision (the object is a vehicle or not) and a mass 0.1 to $\{V, NV\}$.

### B. Spatial and temporal alignment

**Spatial alignment**: Each vehicle detects objects in its local perception's reference frame and builds its dynamic local map. Sensor and vehicle reference frames are the same in the simulator, no need for spatial alignment between these two frames. The frame transformation is done on two levels:

- when a vehicle receives a message, it transforms objects in the global frame.
- when the local map is fused with the distributed map, the local map is transformed in the global frame before combining it with the distributed map.

**Temporal alignment**: We consider in this application that sensors have common time. Sensor detects objects each $\Delta t = 0.1s$. For simplification, we consider that the vehicle will send messages at the same frequency. The transmission time of message is not controlled but it is bounded. It is assumed that data are dated to the time of calculation. Objects are predicted using the function $prediction()$ and fused at the present time. Whatever the time of arrival of the messages, all data is synchronized and processed at the same time. As shown in Algorithm 1, when a vehicle receives a message, it predicts at the reception time its $DDM_r(t)$ and the received message. It fuses these two information at time $t$.

These temporal and spatial alignments are possible with real data by using a GPS pose and time.

### C. Scenario

We have created a scenario with 4 vehicles $\{V_0, V_1, V_2, V_3\}$. Vehicles $\{V_0, V_1, V_2\}$ are equipped with a camera and wifi antenna while $V_3$ is not equipped. Equipped vehicles $(V_0, V_1, V_2)$ can exchange messages. $V_0$ follows $V_1$ for $15s$, $V_3$ overtakes $V_0$ at $1.4s$ and $V_1$ at $3.2s$. $V_2$ at $3s$ comes from the other side. Figure 1 shows the scenario at time $2.7s$. This figure represents the ground truth $(GT)$ of the scenario.

Sensors detect objects each $0.1s$, each vehicle builds its $DLM$, updates its $DDM$ with available data (received messages) and sends its $DPM$ at the same frequency. We tested the scenario for $15s$. Figure 2 shows the result at the instant 2.7s. The first column shows the $DLM$ and the second column presents the $DPM$ (sent distributed map). The field of view of each vehicle detecting the scene is drawn in red.

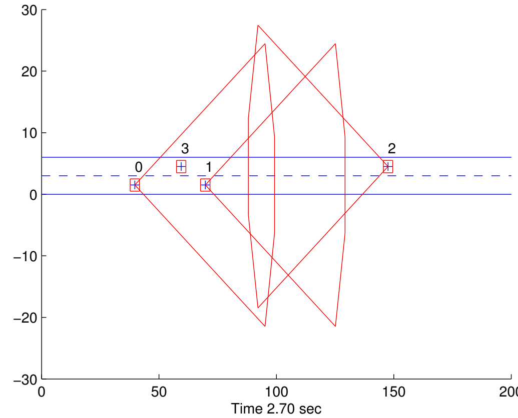


Fig. 1: Scenario at time= 2.7s

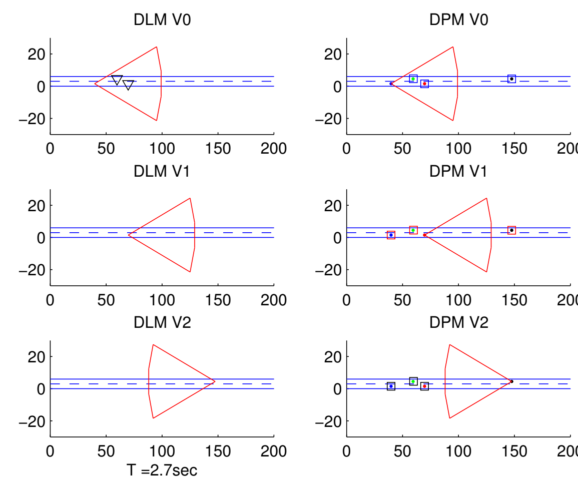


Fig. 2: Scenario at 2.7s: first column is $DLM$ for each vehicle, the second is $DPM$.

**DLM**: $V_0$ detects $V_1$ and $V_3$, this detection is represented by the black triangles in the $DLM$ $V_0$. Vehicles $V_1$ and $V_2$ detect nothing. Vehicles $V_0, V_1, V_2$ communicate with each other.

**DPM**: At each message reception, the receiving vehicle adds the sender vehicle to its list. The $DPM$ of each vehicle is represented by black squares. Vehicles $V_1$ and $V_2$ do not detect other vehicles, but add them when receiving their messages. For example, $V_1$ adds $V_0$ and $V_2$ to its $DDM$ when receiving a message from them, and adds $V_3$ to its list since it was detected by $V_0$. The set of ground truth vehicles are reported as colored dots in the $DPM$, to simplify the verification.

## V. Results

Different tests have been implemented to evaluate the application of the distributed dynamic map. The first test concerns the evolution of the existence mass on a non equipped object. We also show the performances by comparing the rate of true positive $(TP)$, false positive $(FP)$ (non detection) and false negative $(FN)$ (false alarms) of the detections of the three vehicles $V_0$, $V_1$ and $V_2$ in the distributed and local maps. Results are shown in the following.

*a) Evolution of the existence mass:* To make the decision concerning the object existence after updating the distributed map, we calculate the pignistic probability $BetP(NO) = m(NO)+m(\Omega)/2$. This object does not exist if its $BetP(NO)$ is superior to a predefined threshold $\epsilon$. The object is then deleted from the distributed map.

Figure 3 shows the evolution of the existence mass of the vehicle $V_3$ in the local and distributed maps of the vehicles $V_0$, $V_1$ and $V_2$. $V_3$ is not detected at all times by all other vehicles. We should note that in this test $V_0$, $V_1$ and $V_2$ can communicate with each other all the time. The blue curve represents the mass on the existence of the distributed map of each vehicle. $V_0$ detects $V_3$ first at instant 2s. It does not keep it in its distributed map until the pignistic probability reach the threshold $\epsilon$. For this reason, the mass function of $V_0$ does not evolve at time $2.5s$. Afterwards, $V_0$ sends its distributed map to $V_1$ and $V_2$. This explains the appearance of $V_3$ in the distributed map of $V_1$ between 2.5 and $3.8s$ and in the map of $V_2$ between 2.5 and $3s$. At these times, the mass function of the local map is empty (i.e. $m(\Omega) = 1$, $m(V_3) = 0$ and $m(NV_3) = 0$), $V_1$ and $V_2$ didn't detect $V_3$. All vehicles keep $V_3$ in their distributed maps while others detect it. This figure shows the difference between the distributed and local maps. The local map is limited at the vehicle detection while the distributed map increases the field of view of the vehicles.

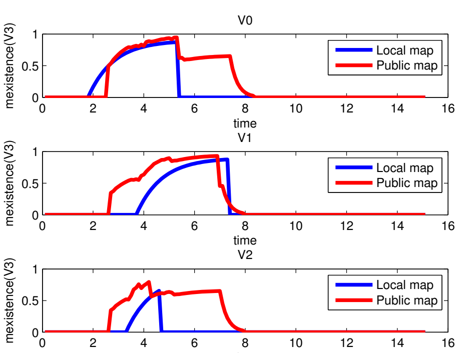


Fig. 3: The existence mass of $V_3$, $m(V_3)$, detected by the others vehicles.

*b) Comparison between DPM and DLM in different situations:* In each situation, we compare the precision and the recall defined as follows:

$$Precision = \frac{TP}{TP+FP}, \; Recall = \frac{TP}{TP+FN}. \tag{7}$$

**Case of a defective sensor:** In this simulation, $V_0$ does not detect $V_3$ between 1.9s and 5.3s. In this case, we search to evaluate the influence of the non detection of a sensor on the vehicle network. We notice the increase of the non detection of $V_0$ in the local map. The effect is more important on the rate of $FP$ of $V_0$. Table I shows a light variation at the level of recall of the $DLM$ and the precision of the $DPM$ of $V_0$.

TABLE I: Precision and recall in the case of defective sensor

| | $V_0$ | | $V_1$ | | $V_2$ | |
|---|---|---|---|---|---|---|
| | Prec | Rec | Prec | Rec | Prec | Rec |
| DLM | 1 | 0.53 | 1 | 0.34 | 1 | 0.34 |
| DPM | 1 | 0.79 | 1 | 0.81 | 0.99 | 0.81 |

**Case of a defective wifi antenna:** This example illustrates the case where $V_2$ has a problem with its wifi antenna. It does not receive messages sent by other vehicles for 7.5s. This problem decreases the good detection of $V_2$ $(VP)$ and increases the rate of non detection $(FP)$. Table II shows the influence of the problem on the recall of $V_2$.

TABLE II: Precision and recall in the case of a defective wifi antenna

| | $V_0$ | | $V_1$ | | $V_2$ | |
|---|---|---|---|---|---|---|
| | Prec | Rec | Prec | Rec | Prec | Rec |
| DLM | 1 | 0.59 | 1 | 0.34 | 1 | 0.34 |
| DPM | 0.93 | 0.73 | 0.97 | 0.77 | 0.97 | 0.57 |

**Change of the range of communication** The change of the range of communication has an influence on the distributed maps of the vehicles. The ranges of communications have been changed according to the following:

- range 1: vehicles have a short range, so that vehicles $V_0$ and $V_1$, which are very close, cannot communicate.
- range 2: the range is increased so that vehicles $V_0$ and $V_1$ can always communicate and $V_3$ can exchange its map when it approaches the other vehicles.
- range 3: the range 3 is bigger than the previous two ranges, which corresponds to the case where all vehicles receive all sent messages.

The table III shows the influence of the change of the range on the performance of the method in this scenario. We notice the increase of the recall of each vehicle when the range increases, as well as a weak variation of the precision. Results of vehicle $V_0$ decrease with range 3 due to false alarms.

TABLE III: Precision and recall in the case of changing the communication range

| | | $V_0$ | | $V_1$ | | $V_2$ | |
|---|---|---|---|---|---|---|---|
| | | Prec | Rec | Prec | Rec | Prec | Rec |
| | range | 1 | 0.59 | 1 | 0.34 | 1 | 0.34 |
| DLM | range 1 | 1 | 0.59 | 1 | 0.34 | 1 | 0.4 |
| DPM | range 2 | 0.98 | 0.83 | 1 | 0.64 | 0.98 | 0.47 |
| | range 3 | 0.94 | 0.79 | 0.98 | 0.81 | 0.99 | 0.83 |

**Send its DLM or its DPM?:** As mentioned in the section I, vehicles can send either their own detections, either the data sent by other vehicles, as received or combined with their own data. In this section, we compare the change of type of the data sent in the messages. Two types of messages are sent: first, vehicles send messages containing their local maps, and then messages containing their distributed maps. The exchange of the distributed maps allows increasing the good detection $VP$ for the vehicles and the field of view of the vehicles. This phenomenon is highlighted in the table IV which shows the increase of the recall in the case of sending a message containing the $DPM$.

TABLE IV: Precision and recall for sending messages containing local map and distributed map

| | | $V_0$ | | $V_1$ | | $V_2$ | |
|---|---|---|---|---|---|---|---|
| | | Prec | Rec | Prec | Rec | Prec | Rec |
| DLM | | 1 | 0.59 | 1 | 0.34 | 1 | 0.34 |
| DPM | Msg DLM | 0.98 | 0.63 | 0.99 | 0.58 | 0.98 | 0.45 |
| | Msg DPM | 0.98 | 0.83 | 1 | 0.64 | 0.98 | 0.47 |

Results are reported on one scenario. These different tests were done on several scenarios, results were similar to those presented here.

## VI. Conclusion

In this paper, we presented a distributed fusion algorithm for distributed dynamic map application. This application is based on the cooperative perception concept, which supposes that vehicles cooperate to improve their field of view. Each vehicle is equipped with sensors allowing it to detect objects in its surroundings, and a wifi antenna in order to communicate with other vehicles. The distributed fusion algorithm allows constructing a dynamic map of the environment, involving objects in the sensor's field of view, as well as the ones sent by other vehicles. The distributed fusion combines the confidence in the existence of objects due to appropriated operators, according to the source of the data. This generates a distributed dynamic map offering an increased perception of the environment. The implementation of such application needs temporal and spatial alignment of the exchanged data, as well as matching of objects by the association algorithm developed in [19]. The distributed dynamic map was validated by simulation implying many vehicles. Future works are concentrated on implementing this approach on the airplug software distribution [1] and validating it on real experimentations data. Results will be reported in future publications.

## ACKNOWLEDGMENT

This work was carried out in the framework of the Labex MS2T, which was funded by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02).

## References

[1] Airplug: https://www.hds.utc.fr/airplug/doku.php.

[2] The safespot project: http://www.safespot-eu.org/.

[3] M. Aeberhard, S. Paul, N. Kaempchen, and T. Bertraml. Object existence probabilty fusion using dempster-shafer theory in a high-level sensor data fusion architecture. In *IEEE Intelligent Vehicles Symposium*, pages 770–775, Baden-Baden,Germany, 05 - 09 June 2011.

[4] J. Curn, D. Marinescu, N. O Hara, , and V. Cahill. Data incest in cooperative localisation with the common past-invariant ensemble kalman filter. In *Proceedings of the 16th Int. Conf. on Information Fusion (FUSION 13)*, Istanbul, Turkey, 09-12 July 2013.

[5] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38:325–339, 1967.

[6] T. Denoeux. Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence. *Artificial Intelligence*, 172:234–264, 2008.

[7] B. Ducourthial, V. Cherfaoui, and T. Denoeux. Self-stabilizing distributed data fusion. *Stabilization, Safety, and Security of Distributed Systems , Lecture Notes in Computer Science*, 7596:148–162, 2012.

[8] A. Houenou, Ph. Bonnifait, V. Cherfaoui, and J.F. Boissou. A track-to-track association method for automotive perception systems. In *IEEE Intelligent Vehicles Symposium*, pages 704–710, Alcala de Henares, Spain, 3-7 June 2012.

[9] N. Karam, F. Chausse, R. Aufrre, and R. Chapuis. Localization of a group of communicating vehicles by state exchange. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2816–2821, Beijing, China, October 9-15 2006.

[10] Seong-Woo Kim, Zhuang Jie Chong, Baoxing Qin, Xiaotong Shen, Zhuoqi Cheng, Wei Liu, and Marcelo H. Ang Jr. Cooperative perception for autonomous vehicle control on the road: Motivation and experimental results. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5059–5066, Tokyo,Japan., 3-7 November 2013.

[11] H. Li and F. Nashashibi. Multi-vehicle cooperative perception and augmented reality for driver assistance: A possibility to see through front vehicle. In *14th International IEEE Conference on Intelligent Transportation Systems*, pages 242–247, Washington, DC, USA, 05-07 October 2011.

[12] H. Li and F. Nashashibi. Cooperative multi-vehicle localization using split covariance intersection filter. In *IEEE Intelligent Vehicles Symposium*, pages 211–216, Madrid,Spain, 03 - 07 June 2012.

[13] A. Martinelli, F. Pont, and R. Siegwart. Multi-robot localization using relative observations. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2808–2813, Barcelona, Spain, April 2005.

[14] L. Merino, F. Caballero, J. M. de Dios, J. Ferruz, and A. Ollero. A cooperative perception system for multiple uav s: Application to automatic detection forest fires. *Journal of Field Robotics*, 23(3-4):165–184, 2006.

[15] M. Munz and K. Dietmayer. Using dempster-shafer-based modeling of object existence evidence in sensor fusion systems for advanced driver assistance systems. In *IEEE Intelligent Vehicles Symposium*, pages 776–781, Baden-Baden,Germany, 05 - 09 June 2011.

[16] A. Rauch, F. Klanner, R. Rasshofer, and K. Dietmayer. Car2x-based perception in a high-level fusion architecture for cooperative perception systems. In *IEEE Intelligent Vehicles Symposium*, pages 270–275, Alcal de Henares, Spain., 3-7 June 2012.

[17] S. Roumeliotis and G. A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.

[18] K. Tischler and B. Hummel. Enhanced environmental perception by inter-vehicle data exchange. In *IEEE Intelligent Vehicles Symposium*, pages 313–318, Las Vegas,USA, 06 - 08 June 2005.

[19] N. El Zoghby, V. Cherfaoui, and T. Denoeux. Optimal object association from pairwise evidential mass functions. In *Proceedings of the 16th Int. Conf. on Information Fusion (FUSION 13)*, pages 351–358, Istanbul, Turkey, 9-12 July 2013.

[20] N. El Zoghby, V. Cherfaoui, B. Ducourthial, and T. Denoeux. Distributed data fusion for detecting sybil attack in vanets. In *Belief Functions: Theory and Applications*, volume 164, pages 351–358, Compiègne, France, 9-11 May 2012.